Université de Montréal
École des Hautes Études Commerciales
École Polytechnique

Université de Montréal
C.P. 6128, Succursale Centre-ville
Montréal (Québec) Canada H3C 3J7

Téléphone : +1 514 343 7575
Télécopieur (fax) : +1 514 343 7121
crt@crt.umontreal.ca
www.crt.umontreal.ca/fr/

CRT-2000-21

Centre de recherche sur les transports (C.R.T.)
Centre for Research on Transportation

NETWORK SYNTHESIS WITH NON-SIMULTANEOUS SINGLE COMMODITY FLOW REQUIREMENTS

by

Michel Gendreau
Martine Labbé
Christelle Wynants

July 2000

101-01786/

PUBLICATION
CRT-2000-21

*NETWORK SYNTHESIS WITH NON-SIMULTANEOUS SINGLE COMMODITY FLOW REQUIREMENTS**

by

Michel Gendreau[1]
Martine Labbé[2]
Christelle Wynants[3]

* *This research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).*

[1] Département d'informatique et de recherche opérationnelle et Centre de recherche sur les transports, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, CANADA H3C 3J7

[2] Service de Mathématiques de la Gestion, Université Libre de Bruxelles, Boulevard du Triomphe C.P. 210/01, 1050 Bruxelles, BELGIUM

[3] Institut de Mathématiques, Université de Liège B37, 12 Grande Traverse, 4000 Sart-Tilman, BELGIUM

Département d'informatique et de recherche opérationnelle – Publication #1177

Dépôt légal – Bibliothèque nationale du Québec,
Bibliothèque nationale du Canada, 2000

ABSTRACT

This paper is dedicated to network synthesis problems with non-simultaneous single commodity flow requirements. An analysis of the complexity of these problems is presented, and new results are proposed. We mainly study the line restoration application since the multi-hour application was already widely covered in the literature. We concentrate on a particular case with uniform costs and requirements. First, we consider continuous capacities and then integer capacities. Some lower bounds and exact solutions for some classes of these problems are provided.

Keywords: Network synthesis, non-simultaneous flows, restoration problem.

RÉSUMÉ

Cet article porte sur les problèmes de synthèse de réseaux avec flux d'un seul produit. Nous analysons la complexité de ces problèmes et présentons de nouveaux résultats. Nous étudions principalement le cas correspondant au rétablissement du service dans un réseau de télécommunications, le cas des réseaux multi-périodes ayant déjà été abondamment traité précédemment. Nous nous concentrons sur les problèmes dans lequels les coûts et les demandes sont uniformes. Nous considérons dans un premier temps le cas de capacités contenues, puis celui de capacités entières. Nous proposons des bornes inférieures et des solutions exactes pour certaines classes de ces problèmes.

Mots-clés: Synthèse de réseau, flux non simultanés, problème de rétablissement du service.

1 Introduction

Network synthesis problems consist in determining the amount of capacity to install on each link of a given graph in order to satisfy some known traffic demands at minimum cost. In this paper, we consider a version where the capacities to be installed need to satisfy several single commodity flow requirements independently from the others. This type of problem is known in the literature as the *network synthesis problem with non-simultaneous single commodity flow requirements* (or $NSSC$ for short).

We distinguish between two applications of this problem arising in telecommunications, namely the *multi-hour* and the *restoration* applications (see Labbé *et al.* [10] for more details about these problems). Given a graph and a set of single commodity flow requirements (i.e. a pair of nodes and a flow value), the multi-hour problem with non-simultaneous single commodity flow requirements (or $MHSC$ for short) consists in determining edge capacities sufficient to route independently each requirement with minimum total cost. Given a network, the line restoration (LR) problem consists in determining edge capacities sufficient to reroute traffic interrupted by any edge failure and with minimum total cost. When an edge fails, the line restoration redirects all the working flow on this edge to alternative routes that connect the two ends of the failed edge. Therefore, to each single edge failure is associated a single commodity flow requirement. As edge failures are assumed to be non-simultaneous, the line restoration defines a $NSSC$ problem. The only difference between the $MHSC$ and LR is that, while all edges are available in $MHSC$, they are not in LR: for each requirement, the failed edge cannot be used.

While most of the literature on $NSSC$ is quite recent, their study was initiated by Gomory and Hu [5] back in 1961. In this paper, they examine a special case of the $MHSC$ problem with fractional capacities and where the basic graph is complete with capacity costs on all edges equal to 1. They propose a polynomial algorithm easy to implement to obtain an optimal solution. Among the recent papers Talluri [15] proposes a new polynomial algorithm and succeeds in reducing the number of edges effectively used. Sridhar and Chandrasekaran [14] extend the polynomial algorithm given by Gomory and Hu [5] for the continuous capacity case to the integer case. In this way, they prove this problem is polynomial. Sakauchi *et al.* [13] consider the LR problem with unit capacity costs. They propose a cutset formulation and solve the linear programming relaxation by iteratively generating constraints. Herzberg [8] extends their work to handle different costs. Grover *et al.* [6] consider the same problem as in [13] with an additional constraint: the traffic on any failed edge can only be rerouted on the k shortest edge disjoint paths (with respect to edge lengths). They develop an efficient heuristic procedure to find feasible integer capacity values. Herzberg and Bye [9] include "hop limit" constraints that impose an upper bound on the number of edges in rerouting paths. In her dissertation, Wang [16] provides a better understanding of the LR problem by studying its polyhedral character-

istics. Strong valid inequalities are identified and used in a branch and cut algorithm (see also Magnanti and Wang [12], Balakrishnan *et al.* [1]) .

In this paper, an analysis of the complexity of $NSSC$ problems is presented, and new results are proposed. We mainly study the LR application, and we consider a particular case with uniform costs and requirements.

The remainder of this paper is organized as follows. Both applications are described and corresponding models are formulated in Section 2. In Section 3, we present some results concerning the complexity of $NSSC$ problems and we study some particular cases. Section 4 is devoted to LR problems on a complete graph with uniform costs and requirements. We successively consider the linear relaxation and the integer case. Section 5 concludes the paper.

2 Problem description

For the LR problem, we consider a telecommunication network on which some demands have been routed. This network can be modeled by an undirected graph $G = (N, E)$ with a set N of nodes and a set E of edges. The failure of an edge interrupts its working flow and creates a demand between its end points. To satisfy this requirement, spare capacities have to be added on the edges at minimum cost. Furthermore, if we assume that two edges never fail simultaneously, then the capacity installed for a given edge failure can be reused for another one.

Let Γ be the set of edges that require protection. For each failed edge $\gamma \in \Gamma$, r^γ denotes the demand on edge γ which corresponds to the given working traffic on edge γ. Since the direction of the working traffic is not important we choose arbitrarily one end node of edge γ as the origin o^γ and the other one as the destination d^γ. Let f_{ij}^γ (respectively f_{ji}^γ) be the amount of restoration flow for the failed edge γ routed on edge $\{i,j\} \neq \gamma$, from node i to node j (respectively from node j to node i). The decision variables y_{ij} represent the amount of spare capacity assigned to edge $\{i,j\}$. Using this notation, we have the following model.

$$\textbf{(LR)} \qquad \min \sum_{\{i,j\} \in E} c_{ij}\, y_{ij} \tag{1}$$

subject to

$$\sum_{j:\{i,j\} \in E} \left(f_{ij}^\gamma - f_{ji}^\gamma\right) = \begin{cases} r^\gamma & \text{if } i = o^\gamma \\ -r^\gamma & \text{if } i = d^\gamma \\ 0 & \text{otherwise} \end{cases} \qquad \gamma \in \Gamma,\ i \in N \tag{2}$$

$$f_{ij}^\gamma + f_{ji}^\gamma \leq y_{ij} \qquad \gamma \in \Gamma,\ \{i,j\} \in E \tag{3}$$

$$f_{ij}^{\gamma} = f_{ji}^{\gamma} = 0 \qquad \gamma \in \Gamma,\ \{i,j\} = \gamma \tag{4}$$

$$f_{ij}^{\gamma}, f_{ji}^{\gamma} \geq 0 \qquad \gamma \in \Gamma,\ \{i,j\} \in E \tag{5}$$

$$y_{ij} \geq 0 \text{ integer}, \quad \{i,j\} \in E \tag{6}$$

This is a mixed integer programming problem. The objective function coefficient c_{ij} in (1) represents the cost per unit of spare capacity installed on edge $\{i,j\}$. We assume that once capacities are installed, the system does not incur any additional routing cost. Constraints (2) are the flow balance constraints that ensure the rerouting of the interrupted flow between nodes o^{γ} and d^{γ} (when edge γ fails), and the flow conservation at each other node. Constraints (3) impose capacity constraints on the edges: the total restoration flow on any edge $\{i,j\}$ (in both directions), when γ fails, cannot exceed the spare capacity installed. Constraints (4) restrict the flows to be zero on the failed edge.

We restrict the capacity design variables y to be non-negative integers and the flow variables f to be non-negative. Given a feasible integer solution y to the problem, we can always find a feasible flow solution that is integer. In fact the latter problem corresponds to a single commodity flow problem for which one can always construct a feasible integer solution whenever the demands and the capacities are integral.

For the $MHSC$ problem, we are given a set Γ of flow requirements, each specified by an origin and destination pair $\{o^{\gamma}, d^{\gamma}\}$, and a corresponding requested flow value r^{γ}. The problem consists in assigning sufficient capacity on each edge of the network to accommodate the flow requirements one at a time, at minimum cost. The formulation $(MHSC)$ is the same than (LR) except that we have to remove constraints (4). Indeed, routing is performed on $G = (N, E)$ for $MHSC$ and on $G^{\gamma} = (N, E \backslash \{\gamma\})$ for LR.

In both cases, we introduce the *requirement graph* $G^r = (N, \Gamma)$. It contains an edge $\{o^{\gamma}, d^{\gamma}\}$ for each requirement $\gamma \in \Gamma$ and we denote by $\bar{G}^r$ its complement (i.e. $\bar{G}^r = K_n \backslash G^r$). Furthermore, we denote by $G^S = (N, S)$ the *solution graph*, where S is the set of edges on which capacity is installed. For LR problems, the solution graph is also called the *reserve network*. In the following, we assume that the requirement graph G^r **is connected** and **spans all nodes in** N. Therefore, the solution graph G^S has to be connected.

3 Problem complexity

We are interested in the complexity of $MHSC$ and LR problems, and some particular cases with uniform requirements (UR) and/or uniform costs UC. First we prove that the (LR) problem is strongly NP-complete by transforming any instance of the Steiner Tree (ST) problem to an instance of the LR problem.

Proposition 1 *The LR problem is strongly NP-complete.*

Proof
We first restate in decision form ST and LR problems.

Steiner Tree problem.
Given an undirected graph $G = (N, E)$, edge costs c_e ($e \in E$), a partition (N_C, N_S) of the node set N and a non negative constant M, does this graph contain a Steiner tree T (i.e. a subtree that includes all the nodes in N_C) such that $\sum_{e \in T} c_e \leq M$?

LR problem.
Given an undirected graph $G = (N, E)$, edge costs c_e ($e \in E$), a set Γ of edges requiring protection, associated demands r^γ and a non negative constant M, do there exist capacities y_e ($e \in E$) and flows f_{ij}^γ ($\gamma \in \Gamma$, $\{i, j\} \in E$) satisfying constraints (2)-(6) of (LR) such that $\sum_{e \in E} c_e y_e \leq M$?

Given any instance of the ST problem, an LR instance can be constructed as follows. Duplicate the compulsory nodes and denote by N_C^* this set of new nodes i^*; i^* corresponding to $i \in N_C$. For each compulsory node i, build an edge $\{i, i^*\}$ of zero cost and denote by E^* this set of edges. Choose arbitrarily one duplicated compulsory node $i_0^* \in N_C^*$ and, for all other duplicated compulsory nodes i^*, build an edge $\{i_0^*, i^*\}$ of cost $(M + 1)$. Denote by E_0^* this set of edges. Consider the graph $G^* = (N \cup N^*, E \cup E^* \cup E_0^*)$ and assume that we need to restore every edge in E_0^* (i.e. $\Gamma = E_0^*$) and that the working flow on each edge is 1, that is $r^\gamma = 1$ for all $\gamma \in \Gamma$. An example of this graph G^* is depicted in Figure 1.

We should have a *Yes* instance for the ST problem if and only if we have a *Yes* instance for the LR problem.
If we have a *Yes* instance for the ST problem, we install one unit of spare capacity on the edges of the tree and on the edges of E^*. Indeed, given a Steiner tree T in G, a solution to the LR problem is obtained by adding all edges of E^* to T and $\sum_{e \in E \cup E^* \cup E_0^*} c_e y_e \leq M$.
Conversely, if we have a *Yes* instance of the LR problem, the solution graph contains a path between i_0^* and any node j^* in $N_C^* \backslash \{i_0^*\}$. Because of the cost structure, such a path between i_0^* and j^* is necessarily of the form $\{i_0^*, i_0, ..., j, j^*\}$ and the intermediate edges are in E. Therefore, this path includes a path in G between i_0 and j. Hence, the resulting reserve network contains a

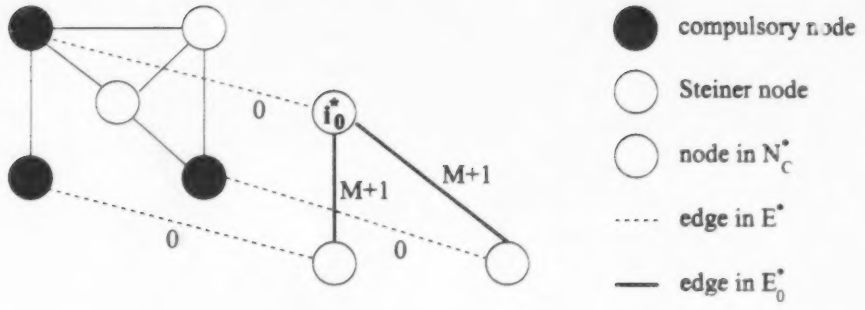


Figure 1: Graph G^*.

connected subgraph which covers all the compulsory nodes; a Steiner tree can thus be extracted from the resulting reserve network, with a total cost lower than or equal to M.

We now prove a stronger result, that is, the strongly NP-completeness of the LR problem. Therefore, for each instance created by our transformation, we have to show that the magnitude of the largest number is bounded by a polynomial of the data length (c.f. Garey and Johnson [4], the similarity assumption). First, note that all the instances of the LR problem created by our transformation have unit demands. It remains to prove that the number M can be bounded by a polynomial of the instance data length. Note that the ST problem remains NP-complete if all edge costs are equal to one (c.f. Garey and Johnson [4] p. 209). Thus, if we consider, in the proof, Steiner tree problems with edge costs equal to one, all the instances of the LR problems created by our transformation verify $\sum_{e \in E} c_e = |E|$ and we can take $M = |E|$. Therefore, the numbers involved in the LR instances created by our transformation are bounded by the data length. Hence, the LR problem is NP-complete in the strong sense. ∎

The NP-completeness of the $MHSC$ problem results from a formulation of the Steiner tree problem in terms of multi-period single flows proposed by Maculan [11]. This formulation shows that ST is a special case of $MHSC$ and is therefore strongly NP-complete. Furthermore, $MHSC$ with uniform costs and uniform requirements is still NP-complete.

The two applications of network synthesis problems with non-simultaneous single commodity flow requirements, namely the $MHSC$ and LR problems, are both NP-complete for arbitrary graphs G. Furthermore, if the graph G is complete, these problems remain NP-complete since any instance of the problem on an arbitrary graph can be transformed into an instance on a complete graph by adding missing edges with a large cost. We can now refine our study to see if the NP-completeness still holds for some particular cases.

We consider the LR problem on a complete graph $G = (N, E)$ with uniform costs $c_e = 1$ $(e \in E)$, and uniform requirements $r^\gamma = r$ $(\gamma \in \Gamma)$. We denote this problem by $ULR(r)$. We prove that the $ULR(r)$ problem is polynomial if $r = 1$ whereas it is NP-complete if $r = 2$. The conclusion is that the $ULR(r)$ problem is NP-complete.

Proposition 2 *The optimal solution of the $ULR(1)$ problem is a Hamiltonian cycle (i.e. a cycle visiting all nodes) or a spanning tree in $\bar{G}^r$.*

Proof
The solution graph G^S has to be connected because we assume that G^r is connected. Therefore the optimal cost is greater than or equal to $n - 1$ $(n = |N|)$. Furthermore, either $\bar{G}^r$ contains a spanning tree, which constitutes an optimal solution graph with edge capacities equal to one, or there is no feasible solution with total cost equal to $n - 1$. In this case, an optimal solution is given by any Hamiltonian cycle in G with a total cost equal to n. ∎

Corollary 1 *The $ULR(1)$ problem is polynomial.*

Lemma 1 *If an $ULR(2)$ instance on a graph $G = (N, E)$ with $n = |N| \geq 3$ admits an optimal solution of cost n, then the solution graph is a Hamiltonian cycle in $\bar{G}^r$ with all edge capacities equal to one.*

Proof
We assume that G^r is connected, $r = 2$, and that the optimal cost is equal to n. Therefore:

(i) $\sum_{j \in N} y_{ij} \geq 2\ , \quad i \in N,$

(ii) the solution graph $G^S = (N, S)$ is connected, thus $|S| \geq n - 1$,

(iii) all edge capacities are equal to 0, 1 or 2.

If there is one edge of capacity 2 , it results from (ii) and (iii) that G^S is a tree with $n - 2$ edges of capacity 1 and one edge of capacity 2. In this case, (i) is not satisfied for at least one leaf of this tree. Consequently $y_e = 1\ \forall\ e \in S$. The solution graph G^S is thus a connected graph with n edges of capacity 1, and then (i) implies that it is a Hamiltonian cycle. ∎

As the solution graph is a Hamiltonian cycle with edge capacity equal to one, this Hamiltonian cycle must belong to $\bar{G}^r$ since the requirement is equal to two.

We prove that the $ULR(2)$ problem is NP-complete by transforming any instance of the Hamiltonian cycle problem (or HC for short) to an $ULR(2)$ instance. An $ULR(2)$ instance is constructed by considering a requirement graph equal to the complement of the original graph. Since we assume that the requirement graph is connected, we have to limit ourselves to graphs whose complement is connected and we thus need the following lemma.

Lemma 2 *The Hamiltonian cycle problem in a graph whose complement is connected is NP-complete.*

Proof
We first state in decision form the general and restricted HC problems.

Hamiltonian cycle problem (HC).
Does a given undirected graph $G = (N, E)$ contain a cycle that visits each node exactly once ?

RestrictedHamiltonian cycle problem (RHC).
Does a given undirected graph $G = (N, E)$, whose complement is connected, contain a cycle that visits each node exactly once ?

Given an HC instance, we check if the complement of G is connected. If it is the case, let $G^* = (N, E)$ (G and G^* are identical). Otherwise, an instance of the Hamiltonian cycle problem in a graph whose complement is connected can be constructed in the following way. As illustrated in Figure 2, a graph $G^* = (N^*, E^*)$ is built from G by splitting each node $i \in N$ into three nodes i', i'', i'''. We denote by N', N'' and N''' the set of nodes i', i'' and i''' respectively.

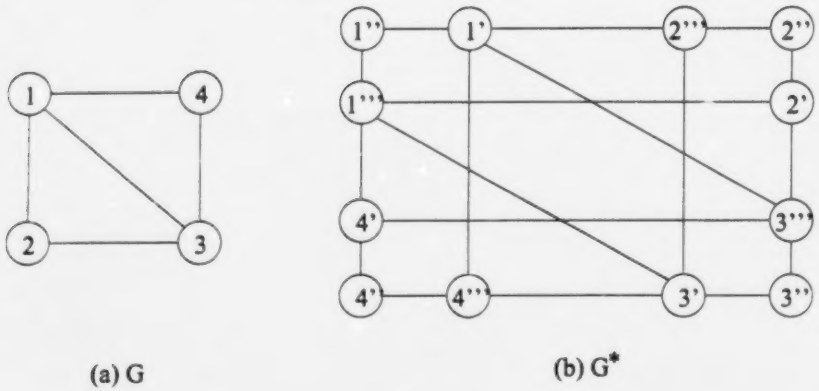


Figure 2: (a) Original graph G (b) Auxiliary graph G^*.

Furthermore, the set E^* is composed of edges $\{i', i''\}$ and $\{i'', i'''\}$ for each node triplet (i', i'', i'''), and $\{i', j'''\}$ and $\{j', i'''\}$ for each edge $\{i, j\}$ in E.

The remaining point consists in showing that $\bar{G}^*$ is connected. On the one hand, notice that there are at most three connected components in $\bar{G}^*$ since nodes in N' are all connected and the same holds for N'' and N'''. On the other hand, vertices i' and i''' are adjacent in $\bar{G}^*$ and the same holds for i' and j'' $(i \neq j)$. Consequently, the graph $\bar{G}^*$ is connected.

We should have a *Yes* instance for the HC problem if and only if we have a *Yes* instance for the RHC problem.
It is trivial if $\bar{G}$ is connected since G and G^* are identical.
Otherwise, if we have a *Yes* instance for HC, G contains a Hamiltonian cycle $(i_1, i_2, ..., i_n)$. Therefore, $(i'_1, i''_1, i'''_1, i'_2, i''_2, i'''_2, ..., i'_n, i''_n, i'''_n)$ is a Hamiltonian cycle in G^*.
Conversely, assume that we have a *Yes* instance for RHC, that is, there is a Hamiltonian cycle in G^*. Note that in any Hamiltonian cycle, nodes i', i'' and i''' are always consecutive. Therefore, by replacing each triplet (i', i'', i''') by the corresponding node i in G, a Hamiltonian cycle in G is obtained. ■

Proposition 3 *The $ULR(2)$ problem is NP-complete.*

Proof
We first state in decision form $ULR(2)$ problem.

$ULR(2)$ problem.
Given a complete undirected graph $G = (N, E)$, edge costs $c_e = 1$ $(e \in E)$, a set Γ of edges requiring protection, associated demands $r^\gamma = 2$ and a non negative constant M, do there exist capacities y_e $(e \in E)$ and flows f_{ij}^γ $(\gamma \in \Gamma,\ \{i, j\} \in E)$ satisfying constraints (2)-(6) of (LR)

such that $\sum_{e \in E} c_e y_e \leq M$?

Given an instance of the RHC problem, an $ULR(2)$ instance is constructed by considering a complete undirected graph with the same node set N, and $G^r = K_n \backslash G$.

If we have a *Yes* instance of the HC problem then $\bar{G}^r = G$ contains a Hamiltonian cycle. Therefore, the solution graph composed of this Hamiltonian cycle with edge capacity equal to one is a feasible solution of the $ULR(2)$ problem with cost $|N|$.
Conversely, if we have a *Yes* instance of the $ULR(2)$ problem, by Lemma 1, the solution graph is a Hamiltonian cycle with edge capacity equal to one. Therefore, this Hamiltonian cycle belongs to $\bar{G}^r = G$. ∎

Table 1 summarizes complexity results. Remember that UR means uniform requirements and UC means uniform costs.

		LR	$MHSC$
Arbitrary graph	General case	NP-complete	NP-complete [11]
	UR and UC	NP-complete	NP-complete [11]
Complete graph	General case	NP-complete	NP-complete [11]
	UR	NP-complete	NP-complete [11]
	UC	NP-complete	Polynomial [14]
	UR and UC	NP-complete	Polynomial
	$r = 1$ and UC	Polynomial	Polynomial

Table 1: Network synthesis problem complexity.

4 Solutions of the $ULR(r)$ problem

In this section, we study the solution structure of $ULR(r)$. In this problem, the unit cost is the same for all the edges and, without loss of generality, we can assume that it is equal to one. Furthermore, requirements induced by single edge failures are uniform, which means that the working traffic on any edge is equal to zero or to a given value r. The cost of any solution is equal to the total amount of spare capacity installed in the network, and we want to minimize this quantity. In the following, we examine both the continuous relaxation and the integer case of the problem.

4.1 Fractional capacities

The continuous relaxation $FULR(r)$ of the $ULR(r)$ formulation, in which the capacity installed on any edge can be fractional, is polynomial since it can be formulated as a linear program. We cannot however use the same arguments as Gomory and Hu [5] for $FUMHSC(r)$ to derive a fast algorithm to solve any instance of this problem. Therefore, we concentrate on subproblems that are interesting in terms of telecommunication networks. Firstly, we construct an interval for the optimal solution value of any $FULR(r)$ instance. We also study the $FULR(r)$ problem with a complete requirement graph and with a requirement graph whose complement contains a Hamiltonian cycle. Secondly, we consider the class of $FULR(r)$ problems with complete multipartite requirement graphs, in particular star graphs, complete bipartite graphs, and complete k-partite graphs. Therefore, we cover the $FULR(r)$ problem for telecommunication networks divided into zones where edges between these zones require protection.

We are given a *complete* undirected graph $G = (N, E)$ and a set of edges Γ which we have to protect. Let A_i be the set of nodes adjacent to node i in G ($A_i = N \backslash \{i\}$), and Γ_i be the set of nodes adjacent to i in the requirement graph G^r. Note that an instance of the $FULR(r)$ problem is exactly defined by its requirement graph and the value of r, and that a solution consists in a vector of capacities $y = (y_e)_{e \in E}$.

We now state, without proof, an interesting scaling property of the $FULR(r)$ problem.

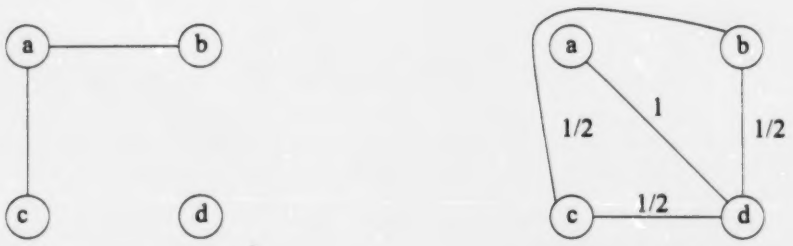
Proposition 4 (Scaling property) *If y^1 is an optimal solution for an $FULR(1)$ instance, then $y^s = sy^1$ is an optimal solution for the corresponding $FULR(s)$ instance.*

We may thus only consider $FULR$ problems with requirements r equal to one (i.e. $FULR(1)$). Furthermore, in the following, we assume that the requirement graph G^r is connected, but this assumption is restrictive. If the requirement graph G^r is not connected, we cannot solve separately the corresponding $FULR$ problems on each connected component. Indeed, we can often build feasible solutions involving several components that are cheaper than the sum of optimal solutions within the components. This is illustrated in the following example.

Let G^r be the graph depicted in Figure 3(a). A feasible solution for this $FULR(1)$ instance is given in Figure 3(b) with a total amount of installed capacity equal to 2.5, while if we solve the $FULR(1)$ problem on the graph composed of the three nodes a, b, and c, the optimal solution is $y_{ab} = y_{ac} = y_{bc} = 1$ for a total cost of $3 > 2.5$.

We now want to construct an interval for the optimal solution value of any $FULR(1)$ instance. The two following lemmas will be used in the construction of this interval.

Lemma 3 *If the requirement graph G^r is complete, that is $\Gamma = E$, then an optimal solution of the $FULR(1)$ problem is given by: $y_e = \frac{1}{n-2} \quad \forall\, e \in E$, and the corresponding cost is $\frac{n(n-1)}{2(n-2)}$.*



(a) Requirement graph G^r (b) Feasible solution

Figure 3: Example: (a) requirement graph (b) feasible solution.

Proof

We first show that this solution is feasible. As illustrated in Figure 4, if any edge $\{a, b\}$ fails, there are $(n-2)$ 2-edge paths from a to b (every other vertex appears once as an intermediate vertex).

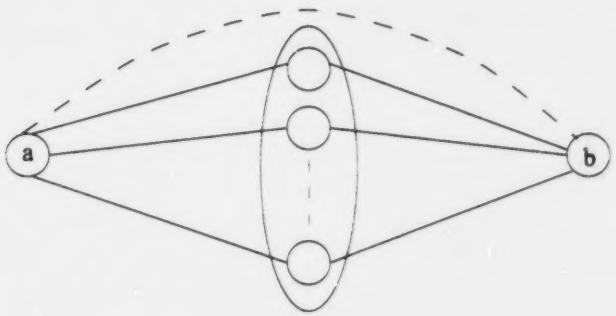


Figure 4: Routing of flow between a and b.

Second, let us show that it is optimal to set $y_e = 1/(n-2)$. One has $A_i = \Gamma_i = N\backslash\{i\}$ for all nodes $i \in N$. It successively follows that, in any feasible solution,

$$\sum_{j \in A_i \backslash \{b\}} y_{ij} \geq 1 , \qquad i \in N,\ b \in \Gamma_i ,$$

$$\sum_{b \in \Gamma_i} \sum_{j \in A_i \backslash \{b\}} y_{ij} \geq |\Gamma_i| = n-1 , \qquad i \in N.$$

Switching the two sums, this reduces to:

$$\sum_{j \in A_i} \sum_{b \in \Gamma_i \backslash \{j\}} y_{ij} \geq n-1 , \ i \in N, \quad \Rightarrow \quad (n-2) \sum_{j \in A_i} y_{ij} \geq n-1 , \ i \in N.$$

Summing over the nodes, one obtains:

$$\sum_{i \in N} \sum_{j \in A_i} y_{ij} \geq \frac{n(n-1)}{(n-2)} \quad \text{or equivalently} \quad \sum_{\{i,j\} \in E} y_{ij} \geq \frac{n(n-1)}{2(n-2)}.$$

As this lower bound is reached when $y_e = 1/(n-2)$, $\forall e \in E$, this solution is optimal. ∎

Lemma 4 *If $\bar{G}^r$ contains a Hamiltonian cycle, an optimal solution of the $FULR(1)$ problem is obtained by installing half a unit of capacity on the edges of this Hamiltonian cycle, and its total cost is $n/2$.*

Proof
This solution is feasible since the Hamiltonian cycle allows the routing of one unit of flow between each pair of nodes. Furthermore, its edges never fail because they belong to $\bar{G}^r$. The optimality results from the fact that $n/2$ is a lower bound. Indeed, as the requirement graph is connected, one has in any feasible solution:

$$\sum_{j \in A_i} y_{ij} \geq 1 , \qquad i \in N \ \Rightarrow \ \sum_{i \in N} \sum_{j \in A_i} y_{ij} \geq n ,$$

or equivalently, $\sum_{\{i,j\} \in E} y_{ij} \geq \frac{n}{2}$. ∎

The following theorem gives an interval for the optimal cost of any $FULR(1)$ instance.

Theorem 1 *For any $FULR(1)$ instance the total amount of spare capacity to install in the network belongs to the interval $\left[\frac{n}{2}, \frac{n(n-1)}{2(n-2)}\right]$.*

Proof
The optimal cost is always greater than or equal to $n/2$ (c.f. the proof of Lemma 4). As the solution proposed in Lemma 3 is feasible for any $FULR(1)$ instance, the optimal cost is lower

than or equal to $\frac{n(n-1)}{2(n-2)}$. ∎

The length of this cost interval is equal to $\frac{n}{2(n-2)}$. For the general $FULR(r)$ problem the corresponding interval is $\left[\frac{nr}{2}, \frac{n(n-1)r}{2(n-2)}\right]$, and the corresponding length is $\frac{nr}{2(n-2)}$. Some intervals are given in Table 2 for different values of n and r.

r	n = 10	n = 20	n = 50	n = 100
1	[5 , 5.625]	[10 , 10.556]	[25 , 25.521]	[50 , 50.510]
2	[10 , 11.250]	[20 , 21.111]	[50 , 51.042]	[100 , 101.020]
5	[25 , 28.125]	[50 , 52.778]	[125 , 127.604]	[250 , 252.551]
10	[50 , 56.250]	[100 , 105.556]	[250 , 255.208]	[500 , 505.102]

Table 2: Solution cost intervals.

We can notice from these results that the intervals for r equal to one are quite tight. Therefore, the solution proposed in Lemma 3 can be considered a good heuristic solution for any $FULR(1)$ instance, that is, for any requirement graph G^r. Furthermore, the relative error decreases as n increases.

A heuristic solution is said to be *asymptotically optimal* when the relative error tends towards 0 if the instance size tends towards infinity.

Theorem 2 *For any $FULR(r)$ instance the solution proposed in Lemma 3 is asymptotically optimal.*

Proof
A lower bound is given by $z_{inf} = nr/2$ and the solution value is $z = \frac{n(n-1)r}{2(n-2)}$.
It follows that:

$$\frac{z - z_{inf}}{z_{inf}} = \frac{2n}{2n(n-2)} = \frac{1}{(n-2)} \rightarrow 0 \quad \text{when } n \rightarrow \infty. \quad \blacksquare$$

In summary, when the requirement graph G^r is complete, an optimal solution graph of the $FULR(1)$ problem is the complete graph with edge capacities equal to $(1/(n-2))$ and this solution is asymptotically optimal for any requirement graph. However, if $\bar{G}^r$ contains a Hamiltonian cycle, then an optimal solution of the $FULR(1)$ problem is obtained by installing half a unit of capacity on the edges of this cycle. Note that if the maximum vertex degree in G^r is lower than or equal to $(\frac{n}{2} - 1)$, then $\bar{G}^r$ contains a Hamiltonian cycle (c.f. Dirac [3]).

The second part of this section is dedicated to the class of $FULR(1)$ problems with complete multipartite requirement graphs.

A graph $G = (N, E)$ is said to be *k-partite* ($k \geq 2$ and integer) if N admits a partition into k classes such that every edge has its end points in different classes. A k-partite graph in which every two vertices from different classes are adjacent is called complete. A complete k-partite graph is denoted by $K_{n_1,...,n_k}$, where n_i is the number of vertices in the i^{th} class.

We consider a complete k-partite graph $K_{n_1,...,n_k}$ with $n_1 = ... = n_k = q$ and we denote it by K_k^q. If G^r is equal to K_k^q, its complement $\bar{G}^r$ is composed of k complete components: $K_1, K_2, ..., K_k$ each containing q nodes. The following explanations give an intuitive idea on how to build a feasible solution.

As G is complete, there are $(k-1)q^2$ edges between one component and the other ones, and they are all in the requirement graph. Therefore, if we take:

$$y_{ij} = \frac{1}{(k-1)q^2 - 1}, \qquad i \in N_r, \ j \in N_s, \ r, s \in \{1, ..., k\} \text{ and } r \neq s,$$

when an edge $\{a, b\}$ ($a \in N_r, b \in N_s$) fails, these capacities ensure that the capacity of the cut formed by N_r (or N_s) and its complement is equal to one. It remains to allocate the spare capacities inside the components. The following proposition gives two solutions.

Proposition 5 *If the requirement graph is a complete k-partite graph K_k^q, an optimal solution of the FULR(1) problem is given by*

$$y_{ij} = \begin{cases} \dfrac{1}{\alpha} & , \ i \in N_r, \ j \in N_s, \ r, s \leq k \text{ and } r \neq s, \\ \dfrac{(k-1)\ q}{\alpha} & , \ i, j \in N_r, \ r \leq k, \end{cases} \tag{7}$$

where $(N_1, ..., N_k)$ is the node set partition defining the k-partite graph K_k^q and $\alpha = (k-1)\ q^2 - 1$. Furthermore, the following solution is also optimal if $q \geq 3$

$$y_{ij} = \begin{cases} \dfrac{1}{\alpha} & , \ i \in N_r, \ j \in N_s, \ r, s \leq k \text{ and } r \neq s, \\ \dfrac{(k-1)\ q\ (q-1)}{2\ \alpha} & , \ \{i, j\} \in C_r, \ r \leq k, \end{cases} \tag{8}$$

where C_r is any cycle connecting all the vertices of N_r.

Proof

To prove the feasibility of the first solution described by equations (7), we assume that an edge

$\{a,b\}$ $(a \in N_r, b \in N_s)$ fails and we route one unit of flow from a to b using the following values:

$$f^{\gamma}_{ai} = \begin{cases} \frac{1}{\alpha} & , i \in N \backslash (N_r \cup \{b\}) \\ \frac{(k-1)q}{\alpha} & , i \in N_r \backslash \{a\} \\ 0 & , i = b \end{cases}$$

$$f^{\gamma}_{ij} = \begin{cases} \frac{1}{\alpha} & , i \in N_r \backslash \{a\},\ j \in \cup_{l \neq r} N_l \\ \frac{1}{\alpha} & , i \in N \backslash (N_r \cup N_s),\ j \in N_s \end{cases}$$

$$f^{\gamma}_{ib} = \frac{(k-1)q}{\alpha}, \quad i \in N_s \backslash \{b\}$$

This routing guarantees the feasibility since it does not overflow the edge capacities and the Flow Balance Constraint (FBC) is satisfied at each node.

To prove optimality, we need a lower bound. As the requirement graph is connected, we have, in any feasible solution

$$\sum_{j \in A_i \backslash \{b\}} y_{ij} \geq 1, \qquad i \in N,\ b \in \Gamma_i .$$

Noting that

$$\Gamma_i = \cup_{l \neq r} N_l, \qquad i \in N_r,\ r \leq k,$$

it follows that

$$\sum_{b \in \Gamma_i} \sum_{j \in A_i \backslash \{b\}} y_{ij} \geq |\Gamma_i| = (k-1)\, q, \qquad i \in N_r,\ r \leq k .$$

As $A_i = (\cup_{l \leq k} N_l) \backslash \{i\}$, we can split up the second sum to obtain

$$\sum_{b \in \Gamma_i} \sum_{j \in N_r \backslash \{i\}} y_{ij} + \sum_{b \in \Gamma_i} \sum_{j \in (\cup_{l \neq r} N_l) \backslash \{b\}} y_{ij} \geq (k-1)\, q, \qquad i \in N_r,\ r \leq k,$$

and therefore,

$$|\Gamma_i| \sum_{j \in N_r \backslash \{i\}} y_{ij} + |\Gamma_i| \sum_{j \in \cup_{l \neq r} N_l} y_{ij} - \sum_{b \in \Gamma_i} y_{ib} \geq (k-1)\, q \qquad \forall\, i \in N_r,\ r \leq k .$$

Merging thetwo first sums into a sum over N and replacing $\sum_{b \in \Gamma_i} y_{ib}$ by the equivalent expression $\sum_{j \in \cup_{l \neq r} N_l} y_{ij}$ and $|\Gamma_i|$ by its value, we get:

$$(k-1)\, q \sum_{j \in N \backslash \{i\}} y_{ij} - \sum_{j \in \cup_{l \neq r} N_l} y_{ij} \geq (k-1)\, q, \qquad i \in N_r,\ r \leq k .$$

Consequently, by summing over the nodes in N_r, it follows

$$(k-1)\,q \sum_{i \in N_r} \sum_{j \in N \setminus \{i\}} y_{ij} - \sum_{i \in N_r} \sum_{j \in \cup_{l \neq r} N_l} y_{ij} \geq (k-1)\,q^2\,, \qquad r \leq k\,. \tag{9}$$

Furthermore, as the requirement graph is K_k^q:

$$\sum_{i \in N_r} \sum_{j \in \cup_{l \neq r} N_l} y_{ij} \;\geq\; 1 + y_{ab}\,, \qquad a \in N_r\,,\; b \in \cup_{l \neq r} N_l\,.$$

Summing over nodes in N_r and over nodes in $(\cup_{l \neq r} N_l)$, it follows

$$(k-1)\,q^2 \sum_{i \in N_r} \sum_{j \in \cup_{l \neq r} N_l} y_{ij} \;\geq\; (k-1)\,q^2 + \sum_{a \in N_r} \sum_{b \in \cup_{l \neq r} N_l} y_{ab}\,,$$

or equivalently,

$$\sum_{i \in N_r} \sum_{j \in \cup_{l \neq r} N_l} y_{ij} \geq \frac{(k-1)\,q^2}{(k-1)\,q^2 - 1}\,. \tag{10}$$

Combining (9) and (10), one obtains

$$(k-1)\,q \sum_{i \in N_r} \sum_{j \in N \setminus \{i\}} y_{ij} \geq \;(k-1)\,q^2 + \tfrac{(k-1)\,q^2}{(k-1)\,q^2-1} \;\geq \frac{(k-1)^2\,q^4}{(k-1)\,q^2 - 1}\,,$$

or equivalently,

$$\sum_{i \in N_r} \sum_{j \in N \setminus \{i\}} y_{ij} \geq \frac{(k-1)\,q^3}{(k-1)\,q^2 - 1}\,, \qquad r \leq k\,.$$

Summing over $r \leq k$, we obtain the following lower bound

$$\sum_{i \in N} \sum_{j \in N \setminus \{i\}} y_{ij} \geq \frac{k\,(k-1)\,q^3}{(k-1)\,q^2 - 1}\,. \tag{11}$$

Therefore, the first solution is optimal since its value reaches this lower bound. Indeed, the total cost of the first solution is

$$\frac{k\,(k-1)\,q^2}{2\alpha} + \frac{k\,(k-1)\,q^2\,(q-1)}{2\alpha} = \frac{k\,(k-1)\,q^3}{2\alpha}\,.$$

We can build another optimal solution by modifying the routing of the flow inside the components. When an edge $\{a, b\}$ $(a \in N_r, b \in N_s)$ fails, the only difference with the routing of the first solution is the routing inside N_r and N_s. Instead of installing spare capacity on all the edges, we use any cycle C_r connecting all the vertices in N_r and any cycle C_s connecting

all the vertices in N_s. If we choose to route one unit of flow from a to b, the source a sends $((k-1)\,q-1)/\alpha$ units of flow out of N_r. The following amount of flow is missing

$$1 - \frac{(k-1)\,q-1}{\alpha} = \frac{(k-1)\,q\,(q-1)}{\alpha}.$$

Furthermore, node a can send some flow on cycle C_r in both directions (clockwise and counter-clockwise). Therefore, cycle C_r must carry at most $\frac{(k-1)\,q\,(q-1)}{2\,\alpha}$ units of flow in each direction, and each node on the cycle different from a will receive $(k-1)\,q/\alpha$ units of flow.

The flow balance constraints are satisfied for each node in $N_r \backslash \{a\}$ since they can send $(k-1)\,q/\alpha$ units of flow out of N_r. The same reasoning holds for nodes in N_s.

Furthermore, the total cost of this solution is

$$\frac{k\,(k-1)\,q^2}{2\,\alpha} + \frac{k\,(k-1)\,q^2\,(q-1)}{2\,\alpha} = \frac{k\,(k-1)\,q^3}{2\,\alpha},$$

which is optimal since it reaches lower bound (11). ∎

Note that the solution graph of the first solution is complete. Therefore, one advantage of the second solution graph is that the number of edges effectively used is lower. Furthermore, if we add some edges to the requirement graph, the components of its complement are no more complete and the first solution is no more feasible. Moreover the second solution is still feasible if the complement of the requirement graph is composed of k connected components of the same dimension each containing a Hamiltonian cycle.

Extending these results to complete k-partite graphs $K_{n_1,\ldots,n_k}$ with $n_r \neq n_s$ $(n, s \leq k)$ remains an open question. However, the following propositions give optimal solutions for a star graph and a complete bipartite graph.

If the requirement graph G^r is a star graph $K_{1,(n-1)}$, that is a graph made up of one node of degree $(n-1)$ and $(n-1)$ nodes of degree one, the solution proposed in Lemma 3 for a complete requirement graph is still optimal. We also give another optimal solution effectively using less edges.

Proposition 6 *If the requirement graph G^r is a star graph, that is,*
$\Gamma = \{\{i_0, i\} : i \in N \backslash \{i_0\}\}$, then an optimal solution of the $FULR(1)$ problem is given by

$$y_e = \frac{1}{n-2}, \qquad e \in E. \tag{12}$$

Furthermore, the following solution is also optimal

$$y_{i_0 i} = 1/(n-2)\,, \qquad i \in N\backslash\{i_0\}, \tag{13}$$
$$y_{ij} = 1/2\,, \qquad \{i,j\} \in C, \tag{14}$$

where C is any cycle that connects all the vertices except i_0.

Proof
First, we consider the solution described by equations (12). Feasibility is immediate since it is the same solution than if G^r is complete. The optimality results from the following lower bound.

If i_0 is the maximum degree node in G^r, as the requirement graph is connected, it follows that in any feasible solution,

$$\sum_{j \in A_{i_0}\backslash\{b\}} y_{i_0 j} \geq 1\,, \qquad b \in \Gamma_{i_0} \quad \Rightarrow \quad \sum_{b \in \Gamma_{i_0}} \sum_{j \in A_{i_0}\backslash\{b\}} y_{i_0 j} \geq |\Gamma_{i_0}| = n-1.$$

Switching the two sums, this successively reduces to

$$\sum_{j \in A_{i_0}} \sum_{b \in \Gamma_{i_0}\backslash\{j\}} y_{i_0 j} \geq n-1 \quad \Rightarrow \quad (n-2) \sum_{j \in A_{i_0}} y_{i_0 j} \geq n-1\,. \tag{15}$$

Furthermore,

$$\sum_{j \in A_i\backslash\{i_0\}} y_{ij} \geq 1\,, \qquad i \in N\backslash\{i_0\} \quad \Rightarrow \quad \sum_{i \neq i_0} \sum_{j \in A_i\backslash\{i_0\}} y_{ij} \geq n-1\,,$$

$$\sum_{\{i,j\} \in E : i,j \neq i_0} y_{ij} \quad \geq \quad \frac{n-1}{2}\,. \tag{16}$$

Consequently, by summing equations (15) and (16), one obtains

$$\sum_{\{i,j\} \in E} y_{ij} \geq \frac{n-1}{n-2} + \frac{n-1}{2} = \frac{n(n-1)}{2(n-2)}.$$

As this lower bound is reached, the solution described by equations (12) is optimal.

The feasibility of the solution given by equations (13) and (14) is trivial, and its total cost is $\frac{n(n-1)}{2(n-2)}$ which is optimal. ∎

Corollary 2 *If the maximum node degree in the requirement graph G^r is equal to $(n-1)$, then an optimal solution of the $FULR(1)$ problem is given by:*
$y_e = \frac{1}{n-2}\,, \qquad e \in E.$

Therefore all requirement graphs obtained by adding some edges to a star graph yield the same optimal solution value and a solution graph that consists in a complete graph with edge capacities equal to $1/(n-2)$.

We now consider a complete bibartite graph K_{n_1,n_2}. A proof similar to the one of Proposition 5 allow us to obtain the following proposition.

Proposition 7 *If the requirement graph is a complete bipartite graph K_{n_1,n_2}, an optimal solution of the $FULR(1)$ problem is given by:*

$$y_{ij} = \begin{cases} \dfrac{1}{n_1 n_2 - 1} & , \ i \in N_1 \ , \ j \in N_2 \ , \\ \dfrac{n_2}{n_1 n_2 - 1} & , \ i,j \in N_1 \ , \\ \dfrac{n_1}{n_1 n_2 - 1} & , \ i,j \in N_2 \ , \end{cases} \tag{17}$$

where (N_1, N_2) is the node partition set defining the bipartite graph K_{n_1,n_2}.
Furthermore, the following solution is also optimal if $n_1, n_2 \geq 3$,

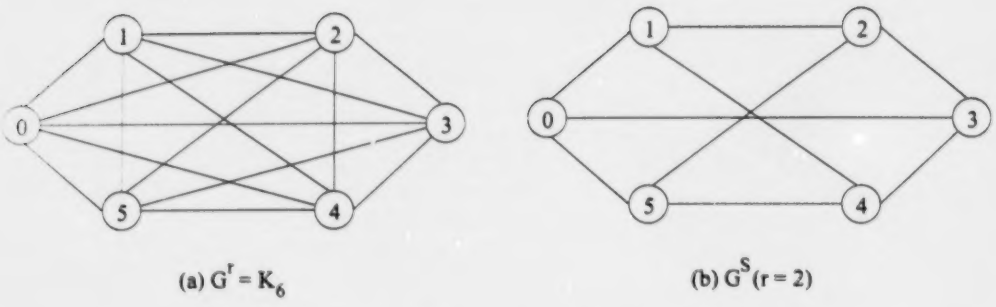
$$y_{ij} = \begin{cases} \dfrac{1}{n_1 n_2 - 1} & , \ i \in N_1 \ , \ j \in N_2 \ , \\ \dfrac{n_1 n_2 - n_2}{2(n_1 n_2 - 1)} & , \ i,j \in C_1 \ , \\ \dfrac{n_1 n_2 - n_1}{2(n_1 n_2 - 1)} & , \ i,j \in C_2 \ , \end{cases} \tag{18}$$

where C_1 (respectively C_2) is any cycle connecting all the vertices of N_1 (respectively N_2).

4.2 Integral capacities

We consider the $ULR(r)$ problem and we assume the requirement graph to be connected. In Section 3 we proved that this problem is NP-complete. In this section we prove that the $ULR(r)$ problem with complete requirement graph is polynomial, and we construct an optimal solution. We first make a few remarks stating some of the characteristics of the problem to highlight its intrinsic difficulty.

The scaling property of the $ULR(r)$ problem with fractional capacities (given in Proposition 4) does not hold for integral capacities. If the capacity vector y^r is an optimal solution for the $ULR(r)$ instance, then $y^{2r} = 2y^r$ is not always an optimal solution for the corresponding $ULR(2r)$ instance. The following example illustrates such a situation. Consider the $ULR(2)$ instance defined by the requirement graph depicted in the Figure 5(a).



(a) $G^r = K_6$ (b) $G^S(r = 2)$

Figure 5: (a) Requirement graph K_6 (b) Solution graph for $r = 2$.

As all the edges are in G^r, and r is equal to 2, any feasible solution y must verify

$$\sum_{j \in N \setminus \{i\}} y_{ij} \geq 3 \ , \qquad i \in N \ , \quad \text{and therefore} \quad \sum_{\{i,j\} \in E} y_{ij} \geq \frac{6 * 3}{2} = 9.$$

The graph depicted in Figure 5(b) with edge capacities equal to 1 is thus an optimal solution graph since its cost value is equal to 9.

If r is equal to 4, a feasible solution is given by $y_{ij} = 1$, $\{i, j\} \in E$. This solution is indeed feasible since each node, in this solution graph, is incident to 5 edges. Therefore if any edge $\{a, b\}$ in K_6 fails, there are four 2-edge paths from a to b. Furthermore, the cost value of this solution is 15, which is lower than $2 * 9 = 18$.

Consequently, the scaling property does not hold for the $ULR(r)$ problem. Note that, as we will see later in this section, the solution proposed for the $ULR(4)$ instance defined by the requirement graph K_6 is optimal.

In Section 3, Lemma 1 proves that if an $ULR(2)$ instance admits an optimal solution of cost n, then the solution graph is a Hamiltonian cycle. This property does not hold for the $ULR(r)$ problem with $r = 2s$ and $s > 1$. Indeed, if an $ULR(2s)$ instance with $s > 1$ admits an optimal solution of cost (ns), the solution graph does not necessary contain a Hamiltonian cycle. This claim is illustrated in the following example.

Consider the $ULR(4)$ instance defined by a requirement graph whose complement is the Petersen graph depicted in Figure 6.

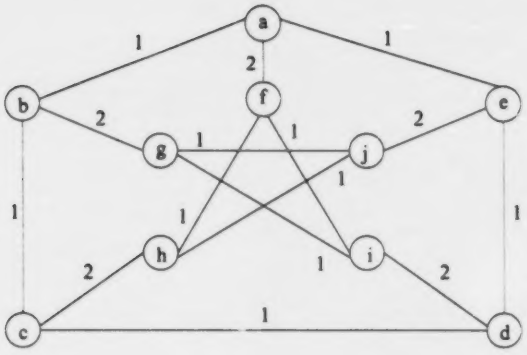


Figure 6: $\bar{G}^r$ = Petersen Graph.

The Petersen graph contains a cycle linking "exterior" nodes: $C_e = (a, b, c, d, e)$ and a cycle linking "interior" nodes: $C_i = (f, h, j, g, i)$. Furthermore, these two cycles are linked by 5 edges: $\{a, f\}$, $\{b, g\}$, $\{c, h\}$, $\{d, i\}$ and $\{e, j\}$.

The capacity values given in Figure 6 define a feasible solution of this $ULR(4)$ instance. Indeed, the exterior and interior cycles can be used in both directions, thus, they can route two units of flow between each pair of their nodes. We consider the three following failures and we detail the routing of 4 units of flow.

- If edge $\{a, c\}$ fails, we can route 2 units of flow from a to c by using cycle C_e in both directions and 2 units of flow can pass through the interior cycle C_i.
- If edge $\{a, g\}$ fails, we can route 2 units of flow from a to b on cycle C_e. These 2 units reach g by using edge $\{b, g\}$. Furthermore, 2 units of flow can be routed on edge $\{a, f\}$ and then on the interior cycle to g.
- If edge $\{f, g\}$ fails, we can route 2 units of flow from f to g on cycle C_i and 2 units of flow can pass through the exterior cycle C_e.

Routings for the other failures can be easily deduced from these ones. Furthermore, the total cost of this solution is 20 which is optimal since,

$$\sum_{q \in N \setminus \{p\}} y_{pq} \geq 4 \ , \quad p \in N \ \text{ which implies } \sum_{\{p,q\} \in E} y_{pq} \geq 20.$$

In summary, this $ULR(4)$ instance has an optimal solution graph equal to the Petersen graph with a total cost equal to 20. However, the Petersen graph is not Hamiltonian (c.f. Bondy [2], p. 54). Consequently, if a $ULR(2s)$ instance with $s > 1$ admits an optimal solution of cost (ns), the solution graph does not necessarily contain a Hamiltonian cycle.

The following proposition provides a lower bound for a $ULR(r)$ instance and a necessary condition to reach this lower bound.

Proposition 8 *The value $\lceil nr/2 \rceil$ is a lower bound on the optimal cost of a $ULR(r)$ instance. Furthermore, if a $ULR(r)$ instance admits an optimal solution of cost $\lceil nr/2 \rceil$, then this solution graph is included in the complement of the requirement graph.*

Proof
As we assume that the requirement graph is connected, one has

$$\sum_{j \in N \setminus \{i\}} y_{ij} \geq r, \qquad i \in N,$$

and consequently,

$$\sum_{i \in N} \sum_{j \in N \setminus \{i\}} y_{ij} \geq nr \quad \text{or equivalently} \quad \sum_{\{i,j\} \in E} y_{ij} \geq \left\lceil \frac{nr}{2} \right\rceil .$$

Furthermore, assume that y^* is an optimal solution of $ULR(r)$ instance with total cost equal to $\lceil nr/2 \rceil$. If edge $\{a, b\}$ is in the requirement graph with $y^*_{ab} > 0$, we have:

$$\sum_{j \in N \setminus \{a\}} y^*_{aj} \geq r+1 , \quad \sum_{j \in N \setminus \{b\}} y^*_{bj} \geq r+1 \text{ and } \sum_{j \in N \setminus \{i\}} y^*_{ij} \geq r , \; i \in N \setminus \{a, b\}.$$

Therefore, we get:

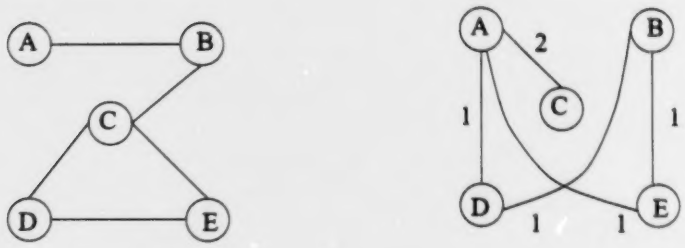
$$\sum_{i \in N} \sum_{j \in N \setminus \{i\}} y^*_{ij} \geq nr+2 \quad \text{or equivalently} \quad \sum_{\{i,j\} \in E} y^*_{ij} \geq \left\lceil \frac{nr}{2} + 1 \right\rceil ,$$

which contradicts the assumptions. ∎

Note that this condition is not sufficient. Consider the $ULR(2)$ instance illustrated in Figure 7(a).

The capacity values given in Figure 7(b) define a feasible solution (cost value equal to 6), and its solution graph in equal to $\bar{G}^r$. This solution is optimal since there is no feasible solution with total cost equal to 5. Furthermore, there is no optimal solution graph containing an edge of G^r. Consequently, the condition given in Proposition 8 is not sufficient to reach the lower bound.

The second part of this section is dedicated to the class of $ULR(r)$ problems with complete requirement graphs. We first propose a better lower bound.



(a) G^r (b) $\bar{G}^r$

Figure 7: A $ULR(2)$ instance: (a) G^r (b) $\bar{G}^r$.

Proposition 9 (Lower bound) *If the requirement graph G^r is complete, then any feasible solution of the $ULR(r)$ problem verifies the following inequalities.*

$$\sum_{j \in N} y_{ij} \geq r+1 , \qquad i \in N \tag{19}$$

$$\sum_{\{i,j\} \in E} y_{ij} \geq \left\lceil \frac{n(r+1)}{2} \right\rceil \tag{20}$$

Proof
As the requirement graph is complete, we have the following inequalities for each node i:

$$\sum_{l \in N} y_{il} \geq r + y_{ij} , \qquad j \in N \backslash \{i\},$$

and the capacity of at least one edge $\{i,j\}$ has to be greater than or equal to one, which proves (19). Summing over nodes in N, we get inequality (20) since capacities are integral. ∎

This lower bound can be used to prove the optimality of the solution we construct. Actually our construction is based on a decomposition of the instances according to the requirement value. We first consider $ULR(n-2)$ problem.

Proposition 10 *If G^r is complete and $r = (n-2)$, then an optimal solution of the $ULR(r)$ problem is given by $y_{ij} = 1$, $\{i,j\} \in E$, and its total cost is equal to $(n(n-1)/2)$.*

Proof
On the one hand, feasibility is trivial. On the other, hand lower bound (19) is reached for each node, hence the optimality is guaranteed. ∎

The following proposition gives an optimal solution of $ULR(r)$ problem with a complete requirement graph and $r < n-2$. This solution depends on the evenness of r and n. In fact, the construction consists in building a Hamiltonian cycle and adding some chords. The aim is to construct a solution graph in which each node is incident to $(r+1)$ edges. Note that, if r is even, we have to add supplementary edges. To simplify the exposition, in the following, vertices are numbered from 0 to $n-1$.

Proposition 11 *If G^r is complete, and if $r < n-2$ then an optimal solution of the $ULR(r)$ problem is given by*

- *if r is odd,*

$$y_{i,i+l} = 1\ , \quad i \in N,\ 1 \leq l \leq \frac{r+1}{2} \tag{21}$$

- *if r is even,*

$$y_{i,i+l} = 1\ , \quad i \in N,\ 1 \leq l \leq \frac{r}{2} \tag{22}$$

and

  - *if n is odd,*

$$y_{i,i+\lfloor \frac{n}{2} \rfloor} = 1\ , \quad 0 \leq i \leq \left\lfloor \frac{n}{2} \right\rfloor - 1 \tag{23}$$

$$y_{\lfloor \frac{n}{2} \rfloor, n-1} = 1 \tag{24}$$

  - *if n is even,*

$$y_{i,i+\frac{n}{2}} = 1\ , \quad 0 \leq i \leq \frac{n}{2} - 1 \tag{25}$$

where addition is taken modulo n.
Furthermore, its total cost is $\left\lceil \frac{n(r+1)}{2} \right\rceil$.

Proof
The feasibility of the solution results from the theorem of Harary (1962 [7]). He considers a graph $H_{m,n}$ which corresponds exactly to the solution graph built in this proposition for $m = r+1$. Harary shows that $H_{m,n}$ is m-connected. Therefore our solution graph is $(r+1)$-connected, that is, there are $(r+1)$ edge disjoint paths between each pair of nodes. Consequently, when one edge fails, r edge disjoint paths remain between its extremities.

The optimality of the solution is derived from lower bounds (19) and (20).

- If r is odd, in the solution graph built in equations (21), each node is incident to exactly $(r+1)$ edges, then lower bound (19) is reached for each node.
- If r and n are even, equations (22) ensure that each node is incident to r edges, and equations (25) add one edge per node; lower bound (19) is thus reached for each node.
- If r is even and n is odd, equations (22) and (23) ensure that each node is incident to $(r+1)$ edges in the solution graph, except the node numbered $n-1$. To guarantee feasibility we have to add one edge (equation (24)) and the node numbered $\lfloor n/2 \rfloor$ becomes incident to $(r+2)$ edges, but lower bound (20) is still reached.

This construction ensures that each node is incident to $(r+1)$ edges, except when r is even and n odd, where one node is incident to $(r+2)$ edges. Therefore the total cost of the solution is $\lceil n(r+1)/2 \rceil$. ■

The solution graph constructed in Proposition 11 is depicted in Figure 8 for a graph with 6 nodes ($r = 2$ and $r = 3$) and in Figure 9 for a graph with 7 nodes ($r = 3$ and $r = 4$).

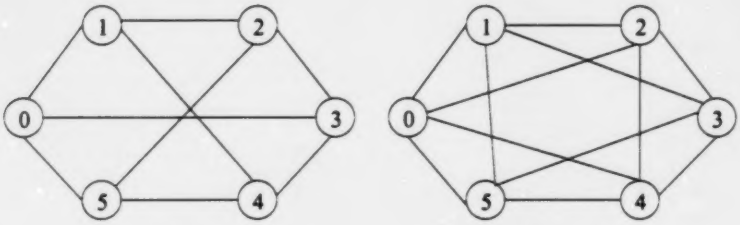


Figure 8: Solution graph for $n = 6$: (a) $r = 2$ (b) $r = 3$.

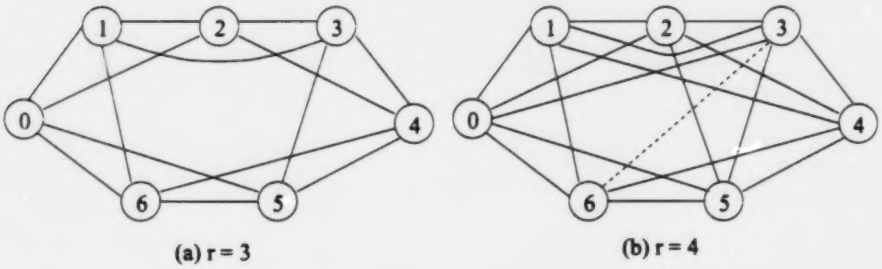


Figure 9: Solution graph for $n = 7$: (a) $r = 3$ (b) $r = 4$.

The last category to consider for the $ULR(r)$ problems with a complete requirement graph concerns requirements greater than $(n-2)$. Let r' and k be the integers such that $r = r' + k(n-2)$

and $0 \leq r' < n-2$. Therefore, $k = \lfloor r/(n-2) \rfloor$.

The basic idea for constructing a solution for the $ULR(r)$ problem consists in summing k optimal solutions of the $ULR(n-2)$ problem with an optimal solution of the $ULR(r')$ problem. In Proposition 13, we prove that this solution is optimal, but before we need to improve lower bound (20).

Proposition 12 (Lower bound) *If the requirement graph G^r is complete, then any feasible solution of one $ULR(r)$ instance verifies the following inequalities :*

(i) if r is a multiple of $n-2$,

$$\sum_{j \in N} y_{ij} \geq r + k \,, \qquad i \in N \,, \tag{26}$$

$$\text{and} \quad \sum_{\{i,j\} \in E} y_{ij} \geq \left\lceil \frac{n(r+k)}{2} \right\rceil \,.$$

(ii) Otherwise,

$$\sum_{j \in N} y_{ij} \geq r + k + 1 \,, \quad i \in N \quad \Rightarrow \quad \sum_{\{i,j\} \in E} y_{ij} \geq \left\lceil \frac{n(r+k+1)}{2} \right\rceil \tag{27}$$

Proof
As the requirement graph is complete, we have the following inequalities for each node i:

$$\sum_{l \in N} y_{il} \geq r + y_{ij} \,, \qquad j \in N \backslash \{i\}.$$

Furthermore, if one edge $\{i,j\}$ fails, r units of flow have to leave node i and $(n-2)$ edges are available. Therefore, for any node i, the capacity of at least one edge $\{i,j\}$ has to be greater than or equal to $r/(n-2)$.

If r is a multiple of $(n-2)$, then

$$\sum_{j \in N} y_{ij} \geq r + \frac{r}{n-2} = r + k \,, \qquad i \in N$$

otherwise,

$$\sum_{j \in N} y_{ij} \geq r + \left\lfloor \frac{r}{n-2} \right\rfloor + 1 = r + k + 1 \,, \qquad i \in N$$

since the capacities are integral. ∎

Proposition 13 *If G^r is complete and $r > (n-2)$, an optimal solution graph of the $ULR(r)$ problem consists in the superposition of a complete graph with edge capacity equal to k and the solution graph of the $ULR(r')$ problem with $r' = r - k(n-2)$ and $k = \lfloor r/(n-2) \rfloor$. Furthermore, the optimal value is $\frac{n(n-1)r}{2(n-2)}$ if r is a multiple of $(n-2)$ and $\left\lceil \frac{n}{2}\left(r + \left\lfloor \frac{r}{n-2} \right\rfloor + 1\right)\right\rceil$ otherwise.*

Proof
As $r = r' + k(n-2)$ with $r' < n-2$, the feasibility results from Propositions 10 and 11, since we superpose feasible solutions. The optimality results from lower bounds (26) and (27). If the requirement r is a multiple of $(n-2)$,

- $r' = 0$ and $k = r/(n-2)$,
- $\sum_{j \in N} y_{ij} = k(n-1)\ , \qquad i \in N,$
- $r + k = (n-2)k + k = (n-1)k,$

lower bound (26) is then reached for all nodes $i \in N$ and the optimal value is:

$$\sum_{\{i,j\} \in E} y_{ij} = \left\lceil \frac{n(r+k)}{2} \right\rceil = \left\lceil \frac{n}{2}\left(r + \frac{r}{n-2}\right)\right\rceil = \frac{n(n-1)r}{2(n-2)}\ .$$

Otherwise,

$$\begin{aligned} \sum_{\{i,j\} \in E} y_{ij} &= \frac{n(n-1)}{2}k + \left\lceil \frac{n(r'+1)}{2} \right\rceil \\ &= \left\lceil \frac{n}{2}\left((n-1)k + r - k(n-2) + 1\right)\right\rceil = \left\lceil \frac{n}{2}(r+k+1)\right\rceil \end{aligned}$$

lower bound (27) is then reached and the optimal value is:

$$\sum_{\{i,j\} \in E} y_{ij} = \left\lceil \frac{n}{2}\left(r + \left\lfloor \frac{r}{n-2} \right\rfloor + 1\right)\right\rceil \qquad \blacksquare$$

Propositions 10, 11 and 13 allow to derive a procedure to solve the $ULR(r)$ problem with a complete requirement graph. This procedure is described in Algorithm 1.

Algorithm 1 ($ULR(r)$ with G^r complete)

- Number vertices from 0 to $n-1$.
- Let k and r' be integers such that $r = r' + k(n-2)$ with $r' < n-2$.
- If $r' = 0$: $y_{ij} = k\ , \qquad \{i,j\} \in E\ ,$
  and the total cost is: $z^* = \frac{n(n-1)k}{2} = \frac{n(n-1)r}{2(n-2)}\ .$

- If $r' \neq 0$ (addition taken modulo n),
  - if r' is odd: $y_{i,i+l} = 1\ , \quad i \in N,\ 1 \leq l \leq \frac{r'+1}{2}$
  - if r' is even: $y_{i,i+l} = 1\ , \quad i \in N,\ 1 \leq l \leq \frac{r'}{2}$
    and
    - if n is odd,

$$y_{\lfloor \frac{n}{2} \rfloor, n-1} = 1 \quad , \quad y_{i,i+\lfloor \frac{n}{2} \rfloor} = 1\ ,\ 0 \leq i \leq \left\lfloor \frac{n}{2} \right\rfloor - 1$$

    - if n is even: $y_{i,i+\frac{n}{2}} = 1\ ,\ 0 \leq i \leq \frac{n}{2} - 1$
  - $y_{ij} \leftarrow y_{ij} + k\ , \quad \{i,j\} \in E\ ,$

and the total cost is: $z^* = \left\lceil \frac{n}{2} \left(r + \left\lfloor \frac{r}{n-2} \right\rfloor + 1 \right) \right\rceil$. □

Corollary 3 *If G^r is complete, the $ULR(r)$ problem is polynomial.*

Results obtained in this section and in Section 4.1 allow us to study the integrality gap for the class of $ULR(r)$ problems with a complete requirement graph. Given a graph and a requirement r, we have the following optimal values.

- Fractional capacities (c.f. Lemma 3): $z_F^* = \frac{n(n-1)r}{2(n-2)}$.
- Integral capacities (c.f. Proposition 13):

$$z_I^* = \begin{cases} \left\lceil \frac{n}{2} \left(r + \left\lfloor \frac{r}{n-2} \right\rfloor + 1 \right) \right\rceil & \text{if } r \text{ is not a multiple of } (n-2), \\ \frac{n(n-1)r}{2(n-2)} & \text{otherwise.} \end{cases}$$

These two costs can be seen as functions of positive integer r. They have the following properties:

- $z_F^*(r) \leq z_I^*(r)\ , \quad r \in I\!N,$
- $z_F^*(r) = z_I^*(r)$ if and only if r is a multiple of $(n-2)$,
- if r is not a multiple of $(n-2)$ (i.e., $r = r' + k(n-2)$ with $0 < r' < n-2$), the integrality gap is:

$$\begin{aligned} z_I^*(r) - z_F^*(r) &= \frac{n(n-1)}{2} \left\lfloor \frac{r}{n-2} \right\rfloor + \left\lceil \frac{n(r'+1)}{2} \right\rceil - \frac{n(n-1)r}{2(n-2)} \\ &\leq \left\lceil \frac{n(r'+1)}{2} \right\rceil \leq \left\lceil \frac{n(n-1)}{2} \right\rceil . \end{aligned}$$

Therefore, the integrality gap is a cyclic function of the requirement r and its period is $(n-2)$.

To terminate this section, note that Algorithm 4.2 can be considered as a heuristic to build feasible solution of a $ULR(r)$ instance with any connected requirement graph. We can study its relative error with respect to the lower bound given in Proposition 8: $z_{inf} = \left\lceil \frac{nr}{2} \right\rceil$.

- If r is a multiple of $(n-2)$ (i.e. $r = k(n-2)$, $r' = 0$): $z_{heur} = \frac{n(n-1)r}{2(n-2)}$, therefore,

$$\frac{z_{heur} - z_{inf}}{z_{inf}} = \frac{\frac{n(n-1)r}{2(n-2)} - \left\lceil \frac{nr}{2} \right\rceil}{\left\lceil \frac{nr}{2} \right\rceil} \leq \frac{n-1}{n-2} - 1 .$$

  This expression tends towards zero when n tends towards infinity.

- If r is not a multiple of $(n-2)$ (i.e. $r = r' + k(n-2)$),

$$z_{heur} = \left\lceil \frac{n}{2} \left(r + \left\lfloor \frac{r}{n-2} \right\rfloor + 1 \right) \right\rceil ,$$

  therefore,

$$\begin{aligned} \frac{z_{heur} - z_{inf}}{z_{inf}} &= \frac{\left\lceil \frac{n}{2} \left(r + \left\lfloor \frac{r}{n-2} \right\rfloor + 1 \right) \right\rceil - \left\lceil \frac{nr}{2} \right\rceil}{\left\lceil \frac{nr}{2} \right\rceil} \\ &\leq \frac{\frac{n}{2} \left(r + \left\lfloor \frac{r}{n-2} \right\rfloor + 1 \right) + 1}{\frac{nr}{2}} - 1 \\ &\leq 1 + \frac{\left\lfloor \frac{r}{n-2} \right\rfloor}{r} + \frac{1}{r} + \frac{2}{nr} - 1 \leq \frac{1}{n-2} + \frac{1}{r} + \frac{2}{nr} \end{aligned}$$

  This expression tends towards $1/r$ when n tends towards infinity.

5 Conclusion

This paper was dedicated to the network synthesis problem with non-simultaneous *single* commodity flow requirements. We mainly studied the line restoration application since the multi-hour application was already widely covered in the literature. We noted that the restoration problem is much more difficult than the multi-hour problem. An analysis of the complexity of these problems was presented, and new results were proposed. We also studied a particular case with uniform costs and requirements. We first considered the linear relaxation (continuous capacities) and then the problem with integer capacities. Some lower bounds and exact solutions for some classes of these problems were provided.

Acknowledgements

We thank the Belgian Commissariat Général aux Relations Internationales (CGRI) and the scientific cooperation program Quebec-Belgian French Community. Financial support for this work was also provided by the Canadian Natural Sciences and Engineering Research Council. Their support is hereby gratefully acknowledged.

References

[1] Balakrishnan, A., T.L. Magnanti, J. Sokol, Y. Wang. 1998. Modeling and Solving the Single Facility Line Restoration Problem. Working paper, *Operations Research Center, MIT, Cambridge.*

[2] Bondy, J.A., U.S.R. Murty. 1976. *Graph Theory with Applications.* The MACMILLAN press LTD.

[3] Dirac, G.A. 1952. Some Theorems on Abstract Graphs. *Proc. London Math. Soc., 2,* 69-81.

[4] Garey, M.R., D.S. Johnson. 1979. *Computers and Intractability, A Guide to the Theory of NP-Completeness.* W.H. Freeman and Company.

[5] Gomory, R.E., T.C. Hu. 1961. Multi-Terminal Network Flows. *SIAM J. Appl. Math. 9,* 551-570.

[6] Grover, W.D., T.D. Bilodeau, B.D. Venables. 1991. Near Optimal Spare Capacity Planning in a Mesh Restorable Network. Proc. *IEEE Globecom,* 2007-2012.

[7] Harary, F. 1962. The Maximum Connectivity of a Graph. *Proc. Nat. Acad. Sci. USA, 48,* 1142-1146.

[8] Herzberg, M. 1993. A Decomposition Approach to Assign Spare Channels in Self-Healing Networks. Proc. *IEEE Globecom,* 1601-1605.

[9] Herzberg, M., S.J. Bye. 1994. Spare Capacity Assignment in Survivable Networks for multi-link and node Failures with Hop Limits. Proc. of the *Sixth International Network Planning Symposium - Networks'94,* Budapest, Hungary, Sept. 4-9, 381-386.

[10] M. Labbé, R. Séguin, P. Soriano, C. Wynants. Network Synthesis with Non-Simultaneous Multicommodity Flow Requirements: Bounds and Heuristics. *ULB-ISRO preprint 99/26,* 1999 (available at http://smg.ulb.ac.be/).

[11] Maculan, N. 1987. The Steiner Problem in Graphs. *Ann. Discr. Math.* 31, 185-212.

[12] Magnanti, T.L., Y. Wang. 1997. Polyhedral Properties of the Network Restoration Problem - with the Convex Hull of a Special Case. Working paper OR 323-97, *Operations Research Center, MIT, Cambridge.*

[13] Sakauchi, H., Y. Nishimura, S. Hasegawa. 1990. A Self-Healing Network with an Economical Spare Channel Assignment. Proc. *IEEE Globecom*, 438-443.

[14] Sridhar, S., R. Chandrasekaran. 1992. Integer Solution to Synthesis of Communication Networks. *Math. Oper. Res. 17 (3).* 581-585.

[15] Tallury, K.T. 1996. Network Synthesis with Few Edges. *Networks*, vol. 27, 109-115 .

[16] Wang, Yi. 1998. *Modeling and Solving the Network Restoration Problem: Single and Multiple Facility Cases.* PhD thesis, MIT, Cambridge.